A Genetic Algorithm for Energy Minimization in Bio-molecular Systems

Xiaochun Weng[1]
Department of Computer Science and Statistics,
University of Rhode Island, Kingston, RI 02881
wengx@cs.uri.edu

Lutz Hamel
Department of Computer Science and Statistics,
University of Rhode Island, Kingston, RI 02881
hamel@cs.uri.edu

Lenore M. Martin
Department of Cell and Molecular Biology,
University of Rhode Island, Kingston, RI 02881
martin@mail.uri.edu

Joan Peckham
Department of Computer Science and Statistics,
University of Rhode Island, Kingston, RI 02881
joan@cs.uri.edu

Abstract Energy minimization algorithms for bio-molecular systems are critical to applications such as the prediction of protein folding. Conventional energy minimization methods such as the steepest descent method and conjugate gradient method suffer from the drawback that they can only locate local energy minima that are extremely dependent on the initial parameter settings of the computation. Here we present an energy minimization algorithm based on genetic algorithms that largely overcomes this drawback of conventional methods because it provides a effective mechanism, through crossover and mutation, to explore new regions of the parameter space without being dependent on a single, preselected parameter setting. This allows the algorithm to cross local energy barriers not surmountable by conventional methods. The algorithm significantly increases the probability of reaching deeper energy minima and locating the global energy minimum. Tests show that the genetic algorithm based approach can achieve much lower final energy than conventional methods. Our genetic algorithm approach differs from other genetic algorithm based approaches in that we do not use the genetic algorithm to directly compute molecular conformations but instead compute a set of parameters to be used in conjunction with a molecular dynamics simulation package (GROMOS96).

1 Introduction

Proteins perform nearly all of the cell's myriad of functions. The multitude of functions proteins perform arises from the huge number of different shapes (conformations) they adopt – structure dictates function. A protein molecule is made from a long polymer chain of a universal set of 20 amino acids, each linked to its neighbor through a covalent peptide bond (proteins are also called polypeptides). Each type of protein encoded by a single gene has a unique sequence of amino acids.

In addition, each type of protein has a particular three-dimensional folded structure that is determined by the linear order of the amino acids in its chain (Alberts et al., 2004). Because long polypeptide chains are very flexible, proteins can in principle fold in an enormous number of ways. Each folded chain is constrained by many different sets of weak non-covalent bonds that form within proteins. These bonds involve atoms in the polypeptide backbone as well as in the amino acid side chains. The non-covalent bonds that help proteins maintain their shape include hydrogen bonds, ionic bonds, van-der-Waals attractions, and the hydrophobicity/hydrophilicity of the side chains. Due to the fact that individual non-covalent bonds are much weaker than covalent bonds, it takes many of these bonds to hold two regions of a polypeptide chain together tightly. The stability of each folded shape will therefore be affected by the combined strength of large numbers of non-covalent bonds.

Protein folding is intimately related to energy minimization. A protein generally folds into the final shape in which the total free energy is minimized, which is the so-called "thermodynamic hypothesis" (Anfinsen, 1973). The fact that a protein can regain the correct conformation on its own indicates that all the information necessary to specify the three-dimensional shape of a protein is contained in its linear amino acid sequence. Misfolded proteins are the origin of a number of serious diseases in animals and human beings. When proteins fold improperly, they can form aggregates that damage cells and even whole tissues. For example, aggregated proteins are the cause of Alzheimer's disease and Huntington's disease. Prion diseases such as the "mad cow disease" are also characterized by changes in protein folding. The prion protein can adopt a special misfolded form that is considered infectious, because it can convert properly folded

[1] Author of correspondence, wengx@cs.uri.edu

proteins into the abnormal conformation. This allows the misfolded prion protein to spread rapidly from cell to cell in the brain, causing the death of the infected animal or human (Alberts et al., 2004).

It is evident that prediction of the three-dimensional conformation that a given protein folds into based on the primary linear sequence of its amino acids is extremely important. Since proteins fold efficiently into a conformation of lowest total energy, all protein folding prediction methods are based on some sort of energy minimization algorithm. Energy minimization algorithms are therefore critical for the computer-based modeling of protein folding.

Protein folding through theoretical simulations faces a variety of significant difficulties. Two of the most challenging problems are the large conformational space that has to be searched and the existence of numerous similar energy minima that hampers conventional energy minimization methods (Hao and Scheraga, 1996). Anfinsen's (1973) thermodynamic hypothesis suggests that protein structures might be predicted from the amino acid sequence by minimizing an appropriate free energy function. Although it has been confirmed in laboratory experiments that the conformations of a correctly folded protein are based on the minimum of the total free energy, a mathematical expression of an energy function over native protein structures that computes the global energy minimum has been difficult to define (Koretke et al., 1998). Therefore, a significant amount of research has been devoted to developing and optimizing simplified energy functions through parametrization (Rosen et al., 2000; Goldstein et al., 1992; Hao and Scheraga, 1996; Seok et al., 2003). Energy function parameter optimization through threading (Goldstein et al., 1992; Maiorov and Crippen, 1994; Thomas and Dill, 1996; Mirny and Shakhnovich, 1996; Koretke et al., 1996) and the lattice models of folding (Shrivastava et al., 1995; Hao and Scheraga, 1996) are two such optimization methods. Another method is the decoy-based parametrization (Seok et al., 2003), in which energy function parameters are determined by maximizing the energy gap between the native protein structures and decoy structures. The above methods are enabled by the assumption that the conformational space is discrete. This restriction is relaxed by another method (Rosen et al., 2000) recently, which can handle energy function parameter optimization for models having continuous degrees of freedom.

We note that in all the above studies the choices of the parameters for the energy functions are experience based, that is, parameters are picked to represent the most reasonable set of initial conditions for the energy minimization function. In very few instances a methodological search over the parameter space is attempted in order to find improved energy minima. In contrast, in the present study we employ a genetic algorithm to search for the energy function parameters, such that the total energy of a bio-molecular system is minimized. We demonstrate that genetic algorithms provide an effective mechanism for overcoming local energy barriers and reaching deeper energy minima. This significantly increases the probability to achieve lower energy values and locating the global energy minimum. Our system uses the GROMOS96 molecular dynamics simulation package (van Gunstern et al, 1996) in order to compute the molecular energies during minimization. Due to this we call our combined system GA-GROMOS. Our system substantially differs from other genetic algorithm approaches, e.g. (T. Dandekar, 1992; S. Schulze-Kremer, 1992), in that we do not directly optimize the conformational structure of the protein but instead we optimize the energy function parameters as embodied by the molecular dynamics package GROMOS96.

2 GA-GROMOS Methodology

We apply genetic algorithms in order to search for parameters that minimize the free energy of a bio-molecular system. The main idea is to encode the simulation parameters and conditions into strings, and apply the genetic algorithm to the strings with an objective function reflecting the magnitude of the system energy. The genetic algorithm guides the search in an informed fashion: good parameters (in terms of achieving lower energy minimum) are retained and exploited to the maximum degree through reproduction, while new regions of the parameter space are explored systematically through crossover and mutation.

We employ the GROMOS96 package (van Gunsteren et al., 1996) to compute the energy of bio-molecular systems. It is worthwhile noting that a single energy minimization computation in GROMOS96 has five distinct phases: The first four stages are molecular dynamics simulations, and the final stage is the energy minimization step using steepest descent or conjugate gradient method. These five stages mimic a process of initialization, heating, constant temperature molecular dynamics simulation, cooling, and energy minimization. GROMOS96 parameters consist of several categories concerning boundary conditions, constraints, potential energy functions, center of mass motions, non-bonded interactions, and program control parameters for these computations. A subset of these parameters is typically selected for optimization and is encoded into genetic algorithm strings. The set of parameters to be optimized is problem dependent, and is chosen based on the physical requirements and configurations of the system.

In the present study we encode the parameters to be optimized into binary strings over the alphabet {0,1}. Translations between genetic algorithm binary strings and values of parameters to be optimized, which can be an integer or a real number, are given by the following rules:

- A binary string of length K is mapped to an integer I with $N_1 \le I \le N_2$ in the following way: the binary string is first converted into a decimal number J ($0 \le J < 2^K$); the decimal number J is then scaled linearly onto the range $[N_1, N_2]$ to obtain the integer I. An integer in the range $[N_1, N_2]$ is encoded into a binary string of length K by reversing the above procedure.
- A binary string of length K is mapped to a real number y with $Y_1 \le y \le Y_2$ in the following way: The binary string is first converted into a decimal number J ($0 \le J < 2^K$); the decimal number J is then scaled linearly onto the range $[Y_1, Y_2]$ to obtain the real number y. A real number in the range $[Y_1, Y_2]$ is encoded into a binary string of length K by reversing the above procedure.

To ensure the accuracy in representing a real number or an integer, the binary string should be sufficiently long. Again, the precise length of the encoding string is determined by the physical requirements of the problem.

The total energy of a bio-molecular system provides a natural measure for the objective function of our genetic algorithm. A lower system energy value should lead to a higher fitness score in our genetic algorithm and vice versa. Furthermore, the fitness score in genetic algorithms is usually required to be non-negative. Taking the above requirements into account, the GROMOS96 energy, which can be positive or negative, is mapped to the genetic algorithm fitness score with the following equation:

$$\textit{Fitness score} = -sign(E)\log_{10}(1+|E|)+G \,. \qquad (1)$$

In the above equation E is the total energy of a bio-molecular system computed by GROMOS96 and $sign(E)$ is a function giving the sign the of the energy. The offset score G (positive constant) ensures the positiveness of the fitness score, and should be set to a sufficiently large value. Once chosen, the value of G remains the same throughout the genetic algorithm computation. If the maximum energy computed by GROMOS96 is $E_{\max} \approx 10^M$ for a system, choosing $G = M + 1.0$ would ensure the positiveness of all the fitness scores.

The GA-GROMOS energy minimization process is illustrated by the flow chart in Figure 1. At the beginning of computation an initial population of random strings is created in the genetic algorithm. We compute the fitness score of each string by decoding the string into GROMOS96 parameters, and running GROMOS96. Based on the fitness score values, the genetic algorithm generates a new population of strings by the rules of reproduction, crossover and mutation (Goldberg, 1989). The system energy and the fitness scores of the new string population are then computed using GROMOS96, and statistics of the string population is collected. If a pre-defined stopping criterion (based on the number of generations or the fitness score values) is satisfied, the computation terminates and the genetic algorithm returns the string with the best fitness score and the corresponding GROMOS96 parameters. Otherwise, the above steps are repeated and the string population is further evolved.

From time to time a GROMOS96 run fails for certain sets of parameter values. The common ones include "shake failure" (Ryckaert et al., 1977) and the "blow-up" of GROMOS96 when the parameters encode illegal initial conditions for the simulation. In these cases the genetic algorithm will assign a pre-defined high energy-value, and thus a very low fitness score, to the present string.

Two types of termination criteria are used in GA-GROMOS. The first criterion is based on the number of generations (referred to as "convergence on generation"). The algorithm terminates when the computation reaches a specified number of generations. The second criterion is based on the convergence of the best fitness score in the population (referred to as "convergence on best score"). If the ratio between the best score of the N-th previous generation, where N is specified by the user, and the best score of the current generation is larger than a specified value α ($0 < \alpha \le 1$), the algorithm terminates.

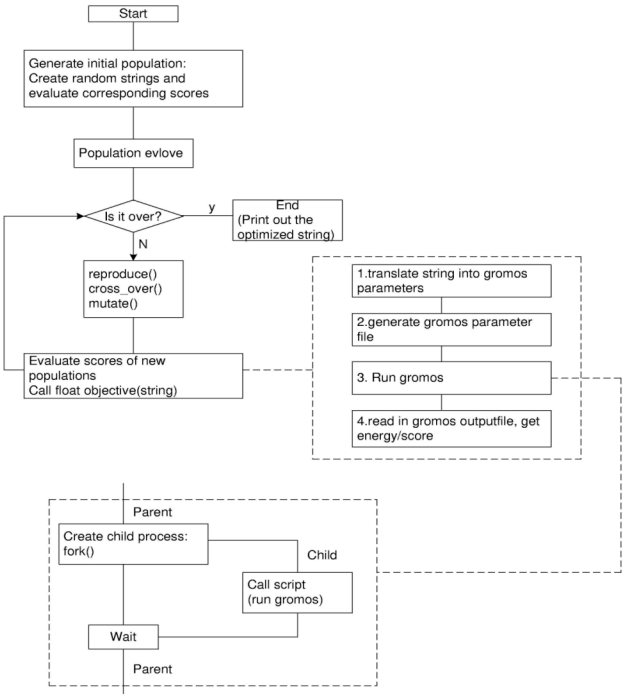

Figure 1. Flow chart for GA-GROMOS energy minimization process.

The GA-GROMOS implementation derives part of its functionality from the MIT GALib C++ library (Wall, 1996). GALib has defined a number of basic classes and functions about strings, population and genetic algorithm, together with a collection of utility classes. To take advantage of the GALib functionalities one needs to define a representation, define the genetic operators, and provide the objective function for computing the fitness scores.

3 Test Molecular System and Results

We tested the above GA-GROMOS algorithm for energy minimization with the following molecular system. The system consists of five amino acids (H-VAL-TYR-ARG-LYS-GLN-O^-, see figure 2 for the chemical structure), one sodium ion (Na^+), three chlorine ions (CL^-), and water as the solvent with 921 water molecules placed in a periodic box with an initial dimension of 3*nm* on each side.

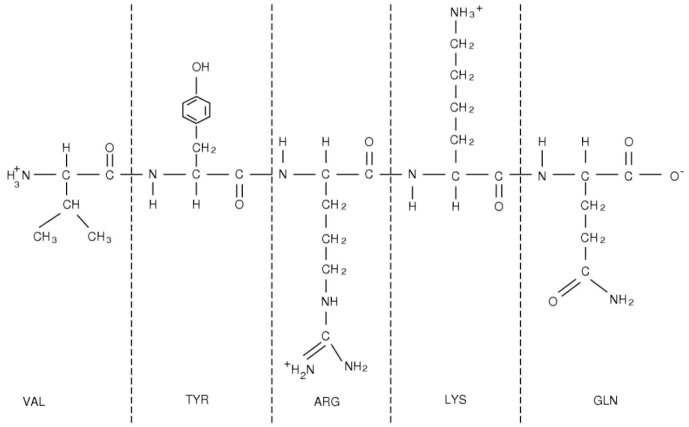

Figure 2. Schematic of test molecule consisting of five amino acids: Valine, Tyrosine, Arginine, Lysine, Glutamine.

Our goal is to minimize the energy of this molecular system and obtain the set of parameters that achieve the lowest energy state and the corresponding conformations. For a given string in a generation and the corresponding set of

parameters translated from this string, we calculate its fitness score by going through the above five GROMOS96 stages and compute the final minimum energy. In total 38 parameters were selected for optimization in this experiment. These include temperature values, the number of time steps computed in each GROMOS96 stage, the temperature at which initial atomic velocities are sampled from a Maxwell distribution, flags controlling which position restraint method is to be used, and "SHAKE" tolerance for solute and solvent. We encode these parameters into a binary string with length 454 bits.

The fitness score of a string is computed from the final minimized system energy of stage five with the following mapping function:

$$fitness = 50 - sign(E)\log_{10}(1+|E|), \quad (2)$$

where E is the final minimized system energy of stage five and G=50. If GROMOS96 encounters a run failure at any stage for a genetic algorithm string, a pre-defined high value is assigned to the system energy and hence a low fitness score to the string. This pre-defined energy has a larger value if the failure occurs at an earlier stage, and a smaller value for a later stage.

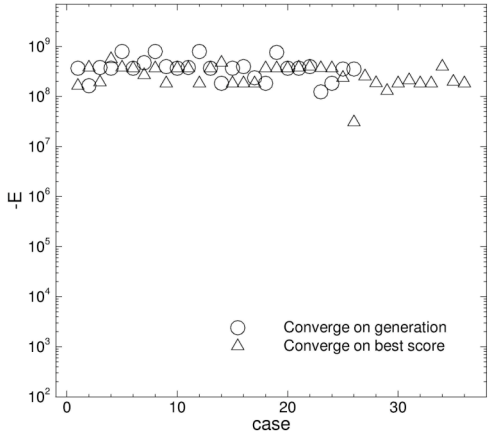

Figure 3 Minimized system energy for all the energy minimization runs with different termination strategies. Note that negative energy is plotted in y-axis.

A total of about 60 independent GA-GROMOS energy minimization runs were performed on the aforementioned molecular system with various crossover probabilities, termination criteria, and population sizes. In Figure 3 we plot the final minimized system energy for all the cases. Note that the negative values of the energy are plotted in the y-axis. Runs with different convergence criteria are denoted by different symbols in the figure. The final minimized system energy reaches the order of magnitude -10^8 in almost all the runs, which is much lower than the minimized energy computed with other methods for comparable molecular systems (Fogh *et al*., 1990). Fogh *et al* report energy minima around -2.1×10^3 kJ/mol. In our experiments, although slight differences exist among the exact minimized energy values from different runs, most runs converge to a minimized energy value around -3.6×10^8 kJ/mol.

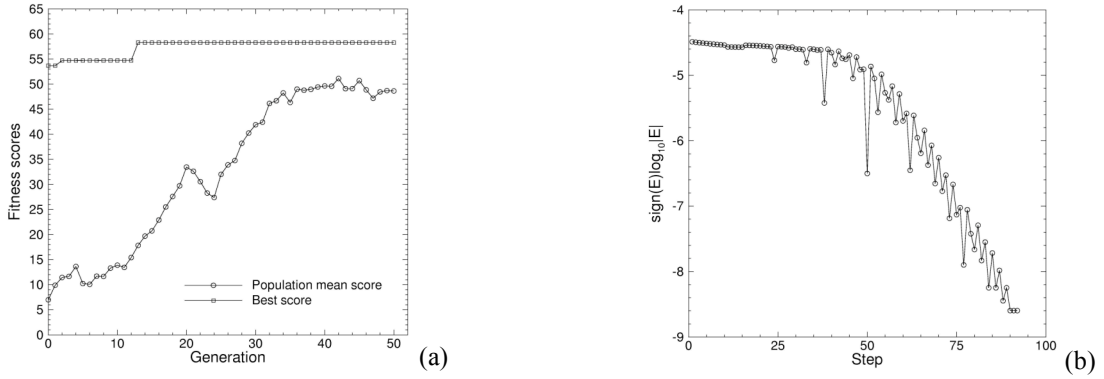

Figure 4. (a) Convergence histories of the best score and population mean score in a typical GA-GROMOS run. The higher the fitness score, the lower the system energy. (b) History of system energy in the energy minimization stage in a typical GA-GROMOS run.

Figure 4(a) shows the history of the best fitness score and the population-mean fitness score as a function of the genetic algorithm generation for a typical GA-GROMOS energy minimization run. The higher the fitness score, the lower the system energy. We first note that both types of fitness scores increase as the genetic algorithm generation increases, indicating that the system energy indeed decreases as the genetic algorithm computation proceeds, although not monotonically with the population mean score. The population best score converges much faster than the population-mean score: the best score reaches the final value at about the fourteenth generation while the mean score reaches the final value only after about the thirty-fourth generation. Figure 4(b) shows the convergence history of the system energy in the energy minimization stage in a typical GA-GROMOS run. The system energy decreases quite slowly initially. After certain steps, an exponential decrease of the system energy is observed, and the system energy is reduced by about four orders of magnitude within about 40 steps. Irregular fluctuations, which are quite large at times, are observed on the energy-step curve, which is indicative of the overshoot-readjustment process when applying the steepest descent or conjugate gradient methods to energy minimizations.

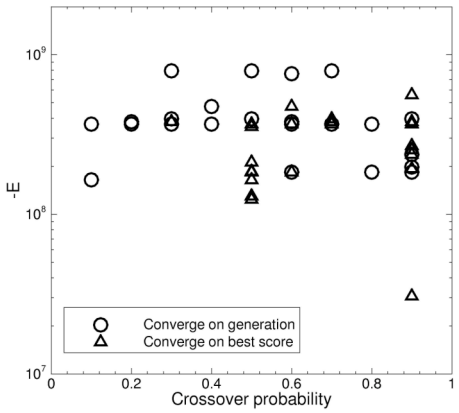

Figure 5. Minimized system energy as a function of cross-over probability, Negative energy is plotted in the y-axis.

We next investigate the effects of several parameters in genetic algorithm on the level of minimized system energy. In Figure 5 we plot the minimized system energy values as a function of the crossover probability in genetic algorithm. Note that the y-axis shows the negative energy values. In theory, if the crossover probability is too low then not enough new regions of the parameter space are explored, and the probability of reaching a low minimized energy is reduced. On the other hand, if the crossover probability is too high a significant number of "good" bit combinations are destroyed. As a result, the probability to reach a lower minimized energy will also decrease. Therefore, to obtain a lower minimized system energy it is conducive to use a crossover probability in the middle. The data in Figure 5 indeed seems to show this trend. The lowest minimized system energy values are realized with crossover probabilities ranging from about 30% to 70%, while too low (such as 10% or 20%) and too high (such as 80% or 90%) crossover probabilities generally yield relatively higher minimized energy values. It might be interesting to investigate why the effect of crossover probabilities are less pronounced in "convergence on best score" experiments. Perhaps the next figure might shed some light on this due to the fact that "convergence on best score" seems to always be outperformed by "convergence on generation".

Figure 6 shows the effect of the termination criteria in genetic algorithm on the final minimized system energy. Two types of convergence criteria are tested: convergence upon the number of generations, and convergence upon the best score. In general, both termination criteria lead to comparable levels of final minimized system energy. However, we observe that the best energy from the criterion based on the number of generations is notably lower than that from the criterion based on the best scores. It is noted that with the criterion based on best scores even for a very high convergence ratio (e.g. 0.9999) the simulation usually terminates after 15 or 20 generations.

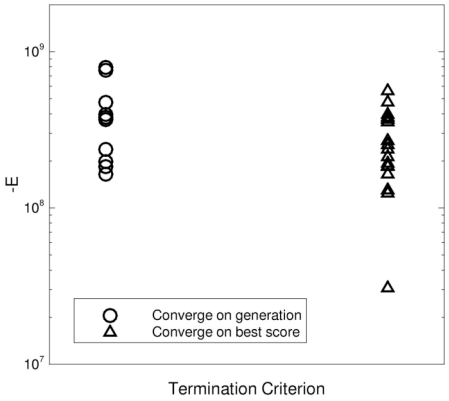

Figure 6. Effect of the termination strategy on the minimized system energy.

In Figure 7 we consider the correlation between the bonds involving hydrogen atoms and the minimized system energy. We observe that, in the final system conformation with the minimized energy, the bonds involving hydrogens have been ignored in the potential energy function in most of the cases. Furthermore, such configurations (with the bonds involving hydrogens ignored) lead to significantly lower minimized system energy compared to configurations taking into account of such bonds. This indicates that the parameter about the bonds involving hydrogens can be essentially set to "ignoring such bonds", and be removed from the set of optimization parameters.

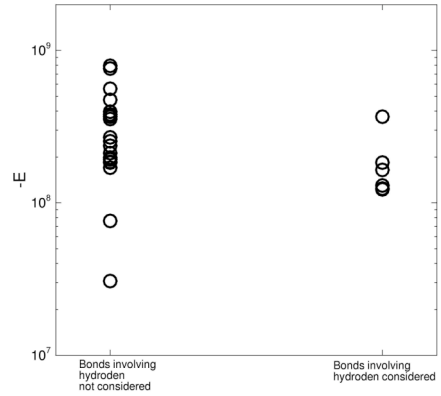

Figure 7. Effect of bonds involving hydrogen on the minimized system energy.

4 Conclusions

Energy minimization methods are essential to applications such as the prediction of protein folding. Energy minimization of bio-molecular systems faces a variety of significant challenges, two of which are the large conformational space that has to be searched and the existence of numerous similar energy minima that hampers conventional energy minimization methods. In this paper we have developed a genetic algorithm – GA-GROMOS – for the energy minimization of bio-molecular systems. Tests have shown that GA-GROMOS algorithm is very effective for energy minimization, achieving significantly lower energy than conventional approaches. The effectiveness of GA-GROMOS lies in the fact that genetic algorithms provide a formal mechanism, through cross-over and mutation, for systematically exploring new regions of the parameter search space and overcoming local energy barriers to locate deeper energy minima, while retaining good parameters through reproduction. It is interesting to note that genetic algorithms coupled with molecular dynamics is more than a method for locating lower energy minima than was otherwise possible and for discovering low-energy system conformations. It is also a way to observe the descent movements of atoms and molecules under certain parameters. One can learn via the genetic algorithm approach how the conditions of the simulation (determined by starting conditions and run parameters) are correlated with the discovery of an energy minimum. Our approach diverges significantly from previous approaches using genetic algorithms for the prediction of protein conformations. In these earlier approaches the genetic algorithm was essentially used to compute conformations that were then tested for the total energy this conformation represented. These approaches did not take advantage of the knowledge embedded in the molecular dynamics simulation packages that have been constructed over the last 3 decades or so. The success of the GA-GROMOS

approach seems to hinge on the fact that we combine two successful paradigms: a powerful search methodology (GA) and an established molecular dynamics simulation package (GROMOS96).

We acknowledge that our test case so far has been a rather small protein; we intend to run our program on a set of larger molecules and verify our results experimentally in the near future.

References:

1. B. Alberts, D. Bray, K. Hopkin, A. Johnson, J. Lewis, M. Raff, K. Roberts and P. Walter, *Essential Cell Biology*, 2nd edition, Garland Science, 2004.
2. C.B. Anfinsen, "Principles that govern the folding of protein chains", *Science*, 181, 223-230, 1973.
3. R.H. Fogh, W.R. Kem and R.S. Norton, "Solution structure of neurotoxin I from the sea anemone Stichodactyla helianthus", *Journal of Biological Chemistry*, 265, 13016-13028, 1990.
4. D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
5. R. A. Goldstein, Z.A. L.-Schulten and P.G. Wolynes, "Protein tertiary structure recognition using optimized Hamiltonians with local interactions", *Proceedings of the National Academy of Sciences USA*, 89, 9029-9033, 1992.
6. M.-H. Hao and H. A. Scheraga, "How optimization of potential functions affects protein folding", *Proceedings of the National Academy of Sciences USA*, 93, 4984-4989, 1996.
7. K.K. Koretke, Z. L.-Schulten and P.G. Wolynes, "Self-consistently optimized energy functions for protein structure prediction by molecular dynamics", *Proceedings of the National Academy of Sciences USA*, 95, 2932-2937, 1998.
8. L.A. Mirny and E.I. Shakhnovich, "How to derive a protein folding potential? A new approach to an old problem", *Journal of Molecular Biology*, 264, 1164-1179, 1996.
9. J.B. Rosen, A.T. Phillips, S.Y. Oh and K.A. Dill, "A method for parameter optimization in computational biology", *Biophysical Journal*, 79, 2818-2824, 2000.
10. J.-P. Ryckaert, G. Ciccotti and H.J.C. Berendsen, "Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular dynamics of n-Alkanes", *Journal of Computational Physics*, 23, 327-341, 1977.
11. C. Seok, J.B. Rosen, J.D. Chodera and K.A. Dill, "MOPED: Method for optimizing physical energy parameters using decoys", *Journal of Computational Chemistry*, 24, 89-97, 2003.
12. I. Shrivastava, S. Vishveshwara, M. Cieplak and A. Maritan, "Lattice model for rapidly folding protein-like heteropolymers", *Proceedings of National Academy of Sciences of USA*, 92, 9206-9209, 1995.
13. P.D. Thomas and K.A. Dill, "An iterative method for extracting energy-like quantities from protein structures", *Proceedings of the National Academy of Sciences of USA*, 93, 11628-11633, 1996.
14. S. Schulze-Kremer, "Genetic Algorithms for Protein Tertiary Structure Prediction", *Parallel Problem Solving from Nature II,* North Holland, pp. 391-400, 1992.
15. T. Dandekar and P. Argos, "Potential of genetic algorithms in protein folding and protein engineering simulations", *Protein Engineering*, 5, 637-645, 1992.
16. W.F. van Gunsteren, S.R. Billeter, A.A. Eising, P.H. Hunenberger, P.K.H. Kruger, A.E. Mark, W.R.P. Scott and I.G. Tironi, *Biomolecular Simulation: The GROMOS96 Manual and User Guide*, Hochschulverlag AG Zurich, 1996.
17. M. Wall, *Galib: A C++ Library of Genetic Algorithm Components*, http://lancet.mit.edu/ga/, 1996.